COMP 308
ARTIFICIAL INTELLIGENCE
PART 3.1 – PROBLEM SOLVING - SEARCHING

Njeri Ireri

Jan – April 2020

We Shall Discuss

- Introduction to problem solving
- Problem Solving Techniques
- Search a s problem solving technique
- Problem Definition
- Search Terminology
- Evaluating a search

Workers are always Searching

Brief info on Ants, http://www.winnipeg-bugline.com/ants.html,

http://www.boston.com/globe/magazine/2002/0623/frontiers.htm

Problem Solving Techniques in A.I

- *Broad* Approaches
  - using search techniques
    - e.g. in Games
  - modeling
    - using Knowledge Base Systems (KBS)
    - using Machine Learning techniques e.g. Artificial Neural Networks, Decision Trees, Case-base reasoning, Genetic algorithms, ..

Searching as a Problem Solving Technique

- **Searching** is the process of looking for the solution of a problem through a set of possibilities (state space)
- **Search** conditions include:
  - Current state – where one is;
  - Goal state – the solution reached; check whether it has been reached;
  - Cost of obtaining the solution
- The **solution** is a path from the current state to the goal state

Searching as a Problem Solving Technique

Process of Searching

- Searching proceeds as follows:
  - Check the current state;
  - Execute allowable actions to move to the next state;
  - Check if the new state is the solution state; if it is not, then the new state becomes the current state and the process is repeated until a solution is found or the state space is exhausted

Search Problem

- The **search problem** consists of finding a **solution plan,** which is a path from the current state to the goal state
- **Representing search problems**
  - A search problem is represented using a directed graph (tree)
  - The states are represented as nodes while the allowed steps or actions are represented as arcs (branches)
- **A search problem is defined by specifying:**
  - State space;
  - Start node;
  - Goal condition, and a test to check whether the goal condition is met;
  - Rules giving how to change states
  - Path cost

Problem Definition - Example, 8 puzzle

5	4	
6	1	8
7	3	2

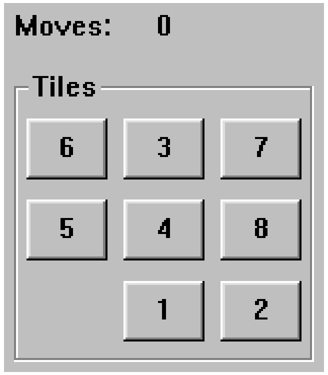
Initial State

1	4	7
2	5	8
3	6	

Goal State

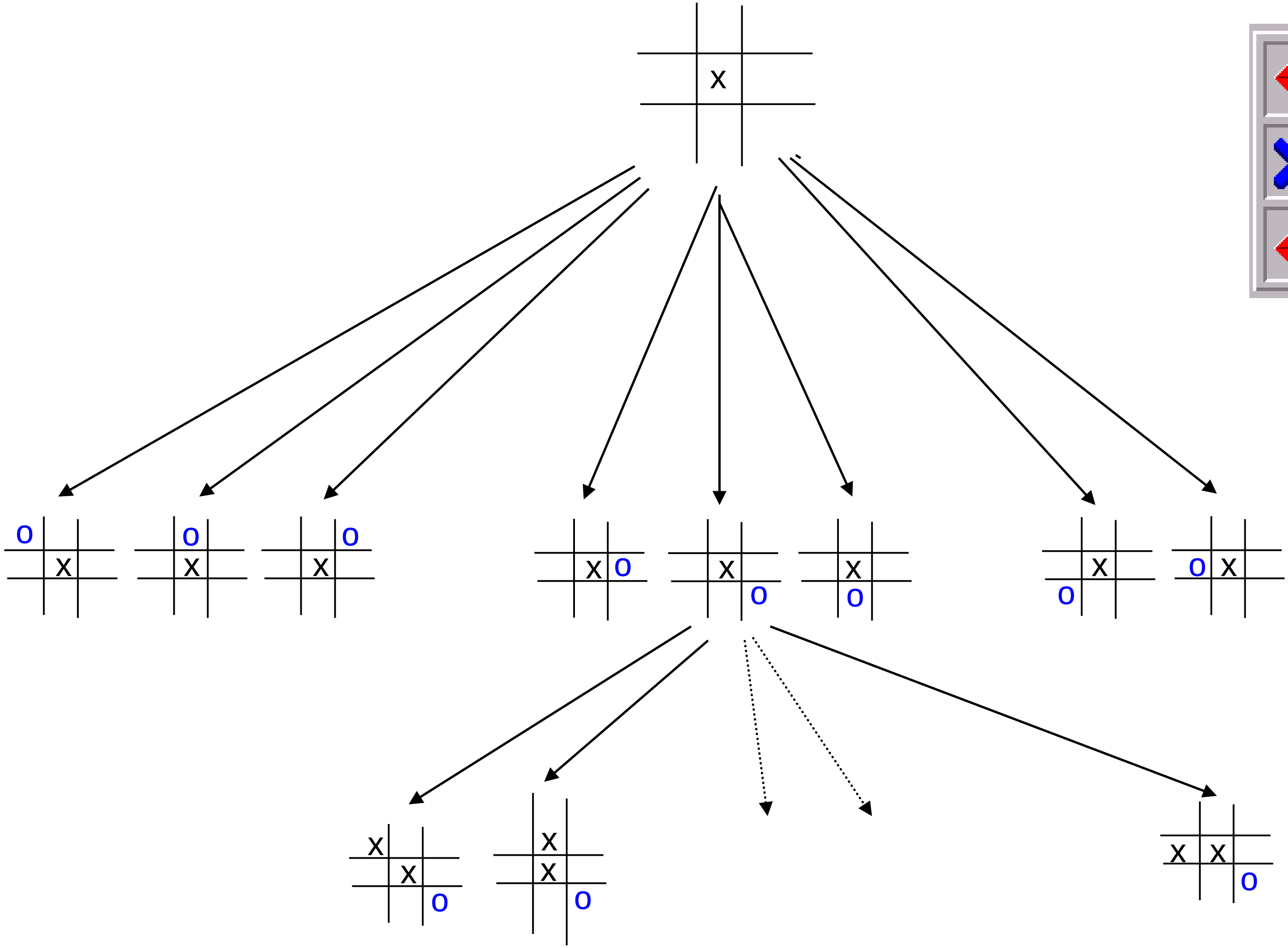
Problem Definition - Example, 8 puzzle

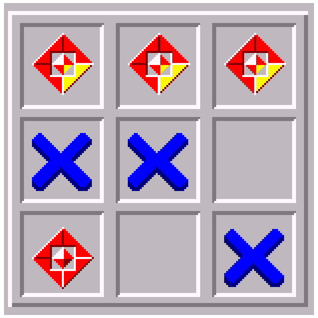
- States
  - A description of each of the eight tiles in each location that it can occupy. It is also useful to include the blank



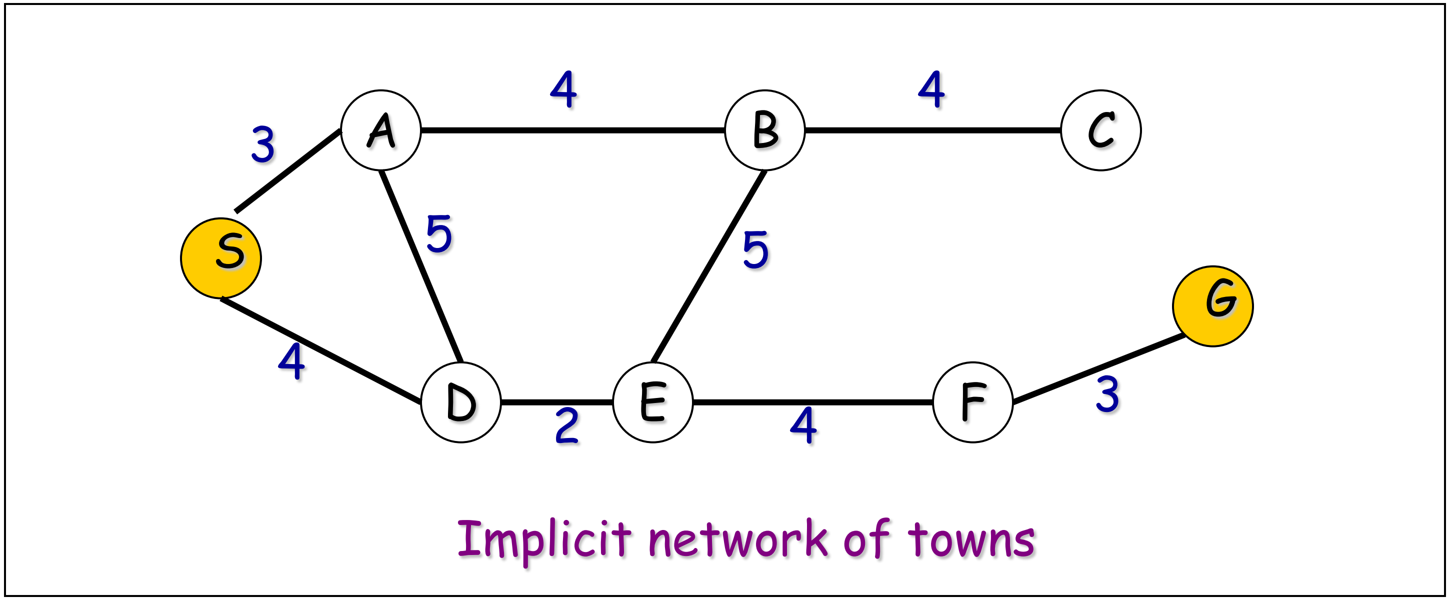
- Operators/Action
  - The blank moves left, right, up or down
- Goal Test
  - The current state matches a certain state (e.g. one of the ones shown on previous slide)
- Path Cost
  - Each move of the blank costs 1

Problem Definition - Example, tic-tac-toe



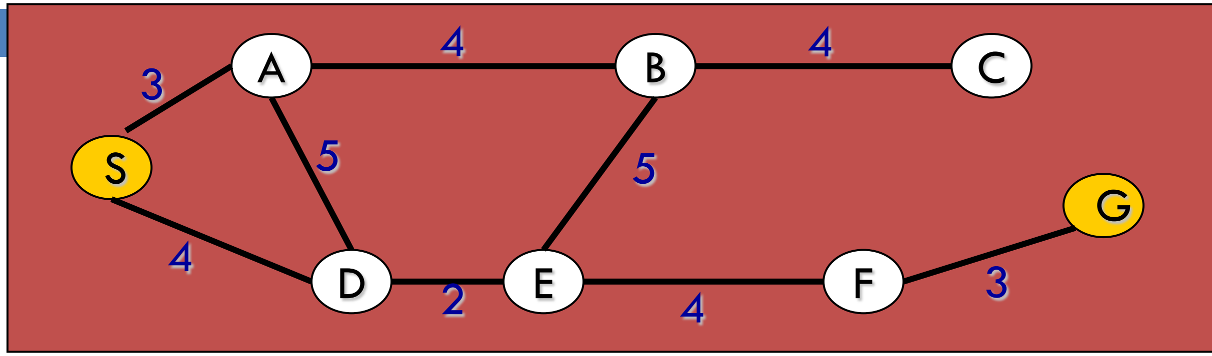


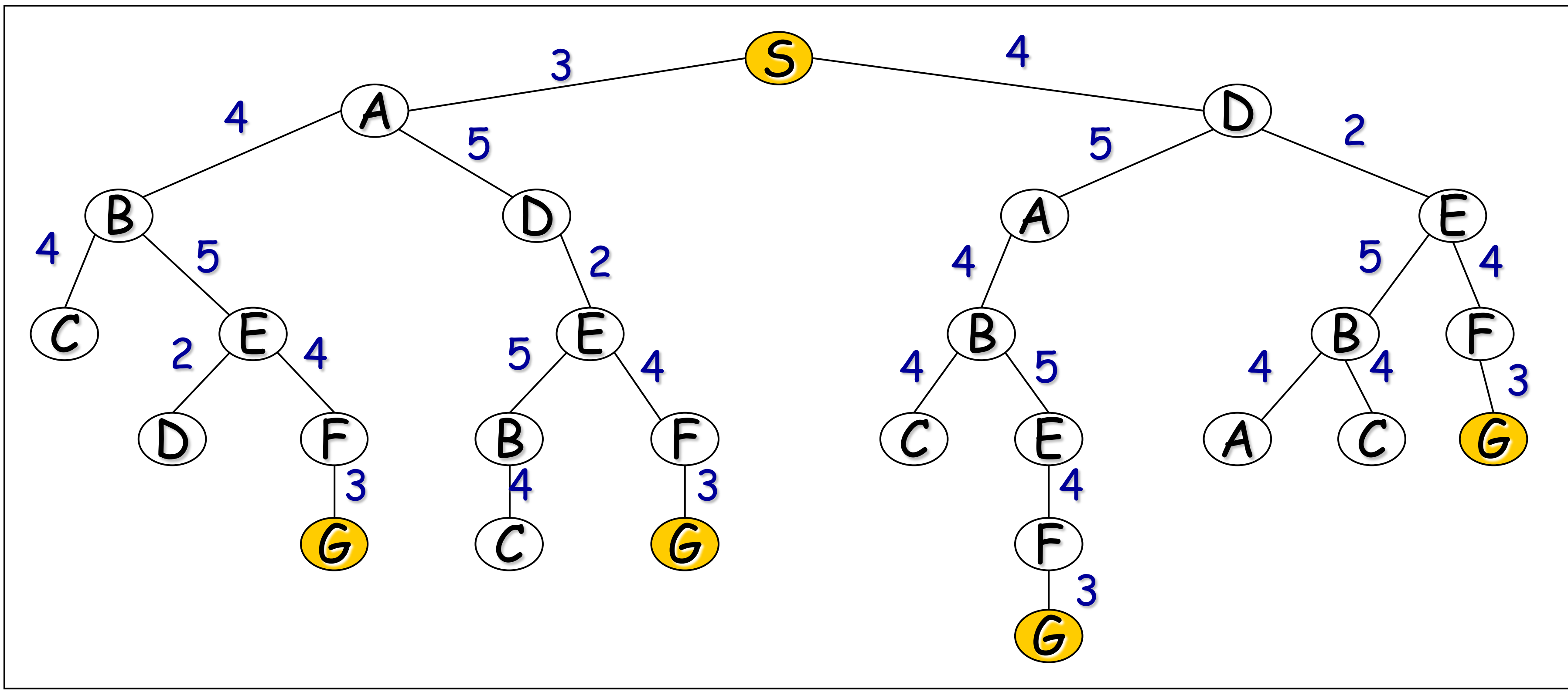
Tree/Path Example:



- Two possible tasks:
  - 1. FIND a (the) path. = computational cost
  - 2. TRAVERSE the path. = travel cost
- 2. relates to finding optimal paths

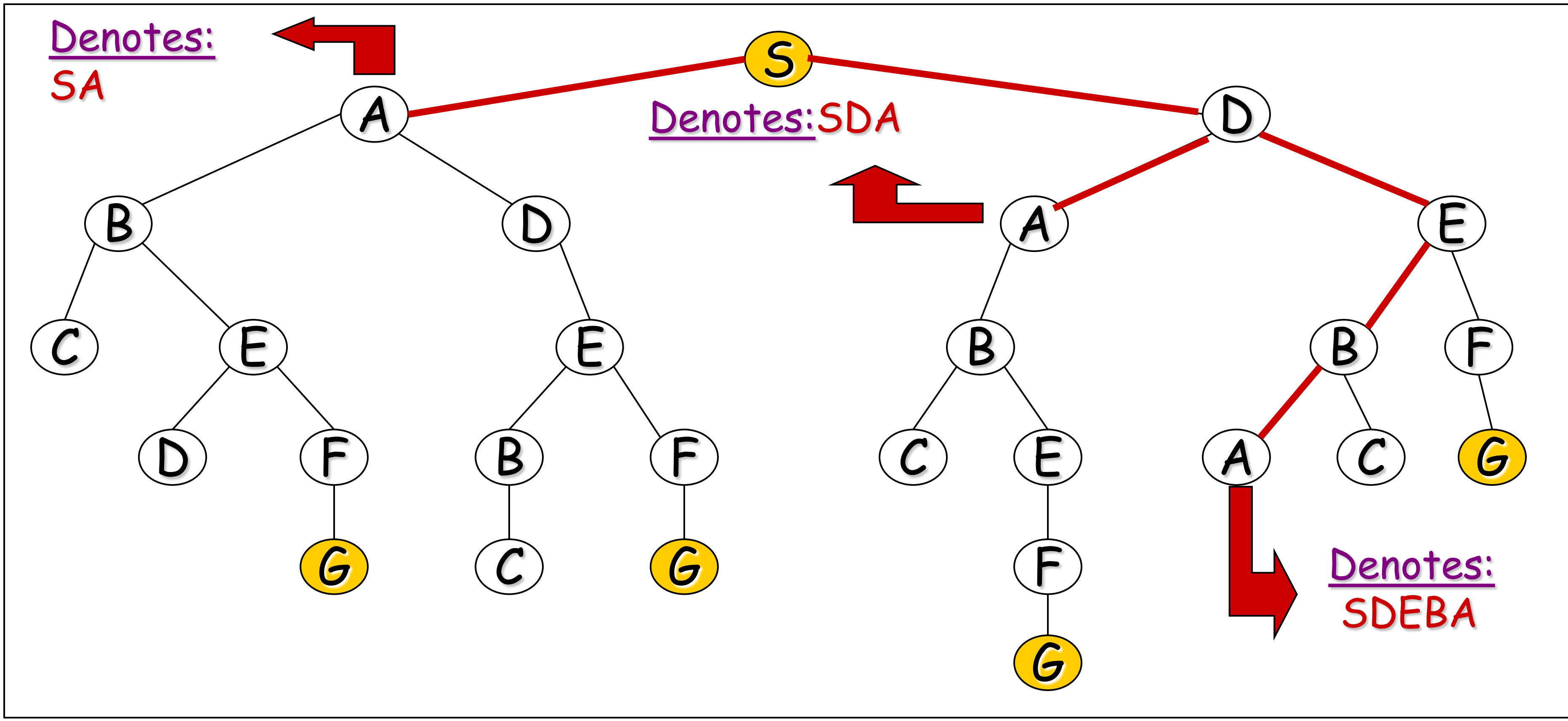
The associated loop-free tree of partial paths





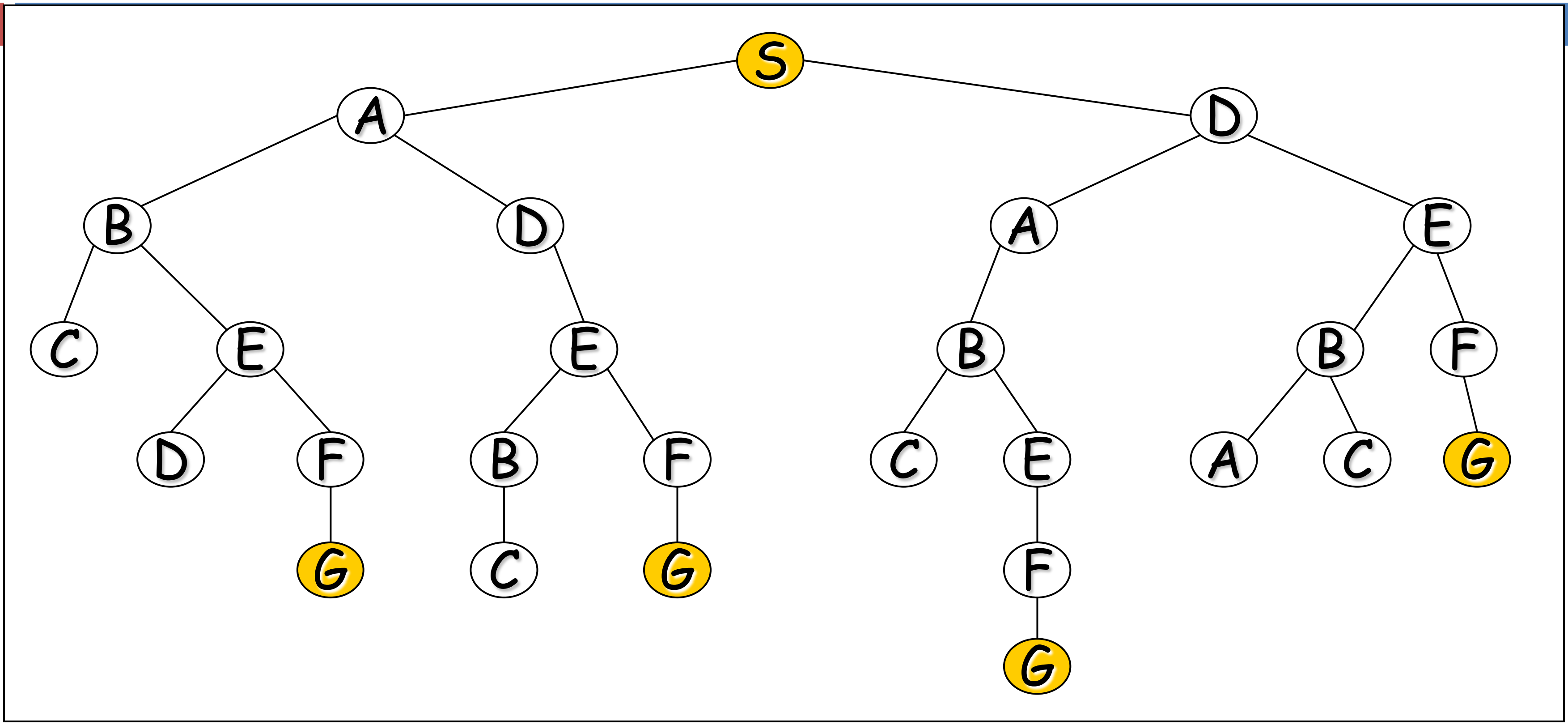
Paths:

- We are not interested in optimal paths here, so we can drop the costs



Note: Nodes do not denote themselves, but denote the partial path from the root to themselves!!

Terminology:



- Node, link (or edge), branch, arc
- Parent, child, ancestor, descendant
- Root node, goal node
- Expand / Open node / Closed node / Branching factor

Using a Tree – The Obvious Solution?

- **But**
  - It can be wasteful on space
  - It can be difficult to implement, particularly if there are varying number of children (as in tic-tac-toe)
  - It is not always obvious which node to expand next
    - We may have to search the tree looking for the best **leaf node** (sometimes called the **fringe** or **frontier** nodes). This can obviously be computationally expensive

How Good is a Solution?

- Does our search method actually find a solution?
- Is it a good solution?
  - Path Cost
  - Search Cost (Time and Memory)
- Does it find the optimal solution?
  - But what is optimal?

Evaluating a Search

- Completeness
  - Is the strategy guaranteed to find a solution?
- Time Complexity
  - How long does it take to find a solution?
- Space Complexity
  - How much memory does it take to perform the search?
- Optimality
  - Does the strategy find the optimal solution where there are several solutions?

Search Trees

- Some issues:
  - Search trees grow very quickly
  - The size of the search tree is governed by the branching factor
  - Even this simple game tic-tac-toe has a complete search tree of 984,410 potential nodes
  - The search tree for chess has a branching factor of about 35

Exercise

1. How are problems solved in artificial intelligence?
2. What is searching?
3. (a) What are the things that could specify a search problem?

   (b) Supposing you had a robot that is supposed to maneuver it self on a factory floor cluttered with numerous machines and boxes containing both raw and finished materials from the back to the front of the factory. What would be specified for the case in (a)

Exercise

4. (a) Playing the 8 Puzzle game, draw a search tree to level three for the initial game state given in figure 1

   (b) How many moves would you require to complete the game given in figure 1

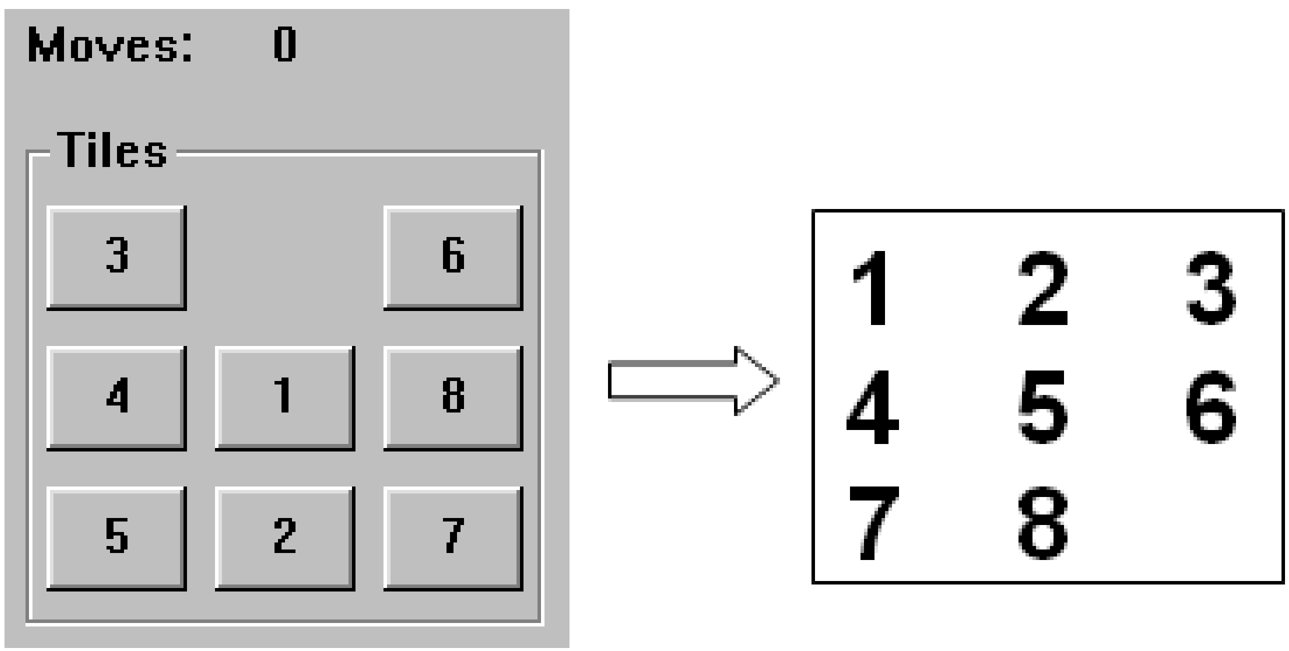


Figure 1. 8 Puzzle (Left-Start State and Right-Goal State)